

FIG. 5

THE JOURNAL OF THE

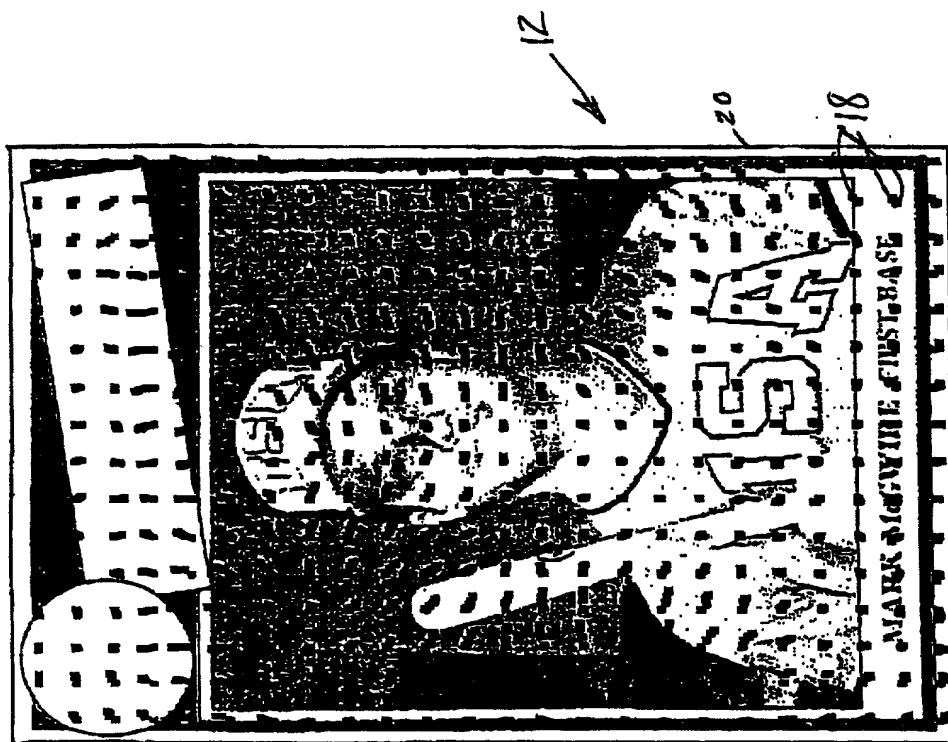


FIG. 6

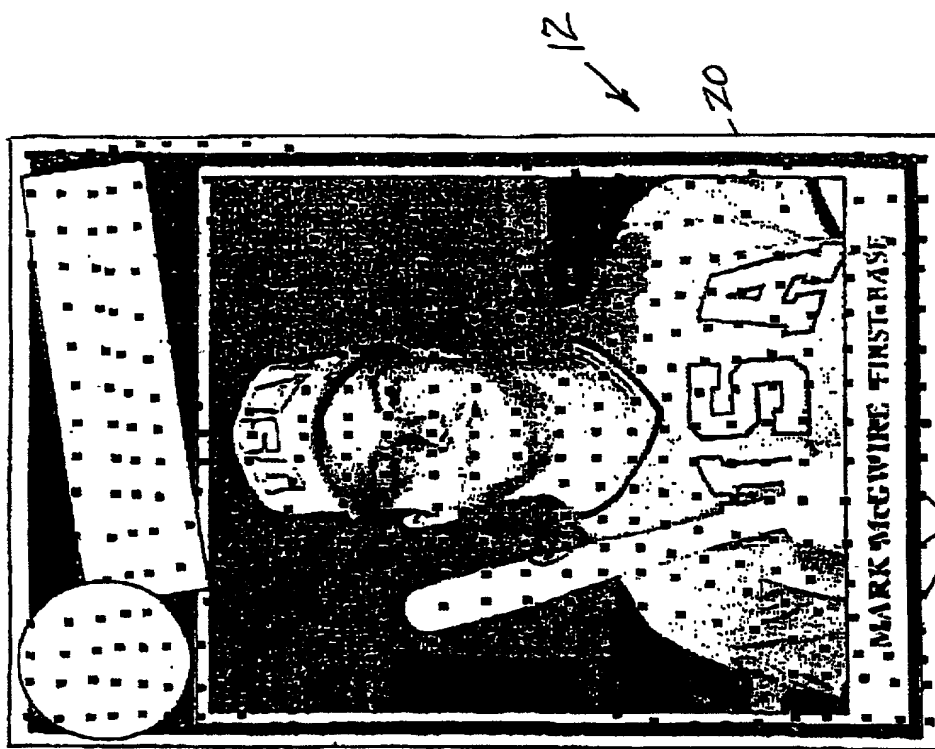


FIG. 7

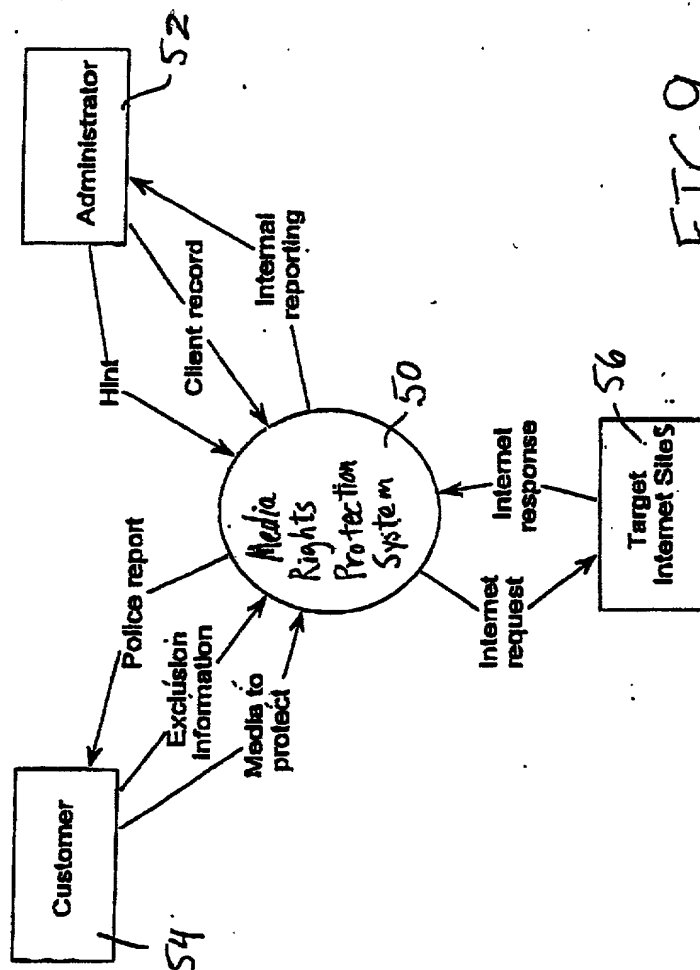
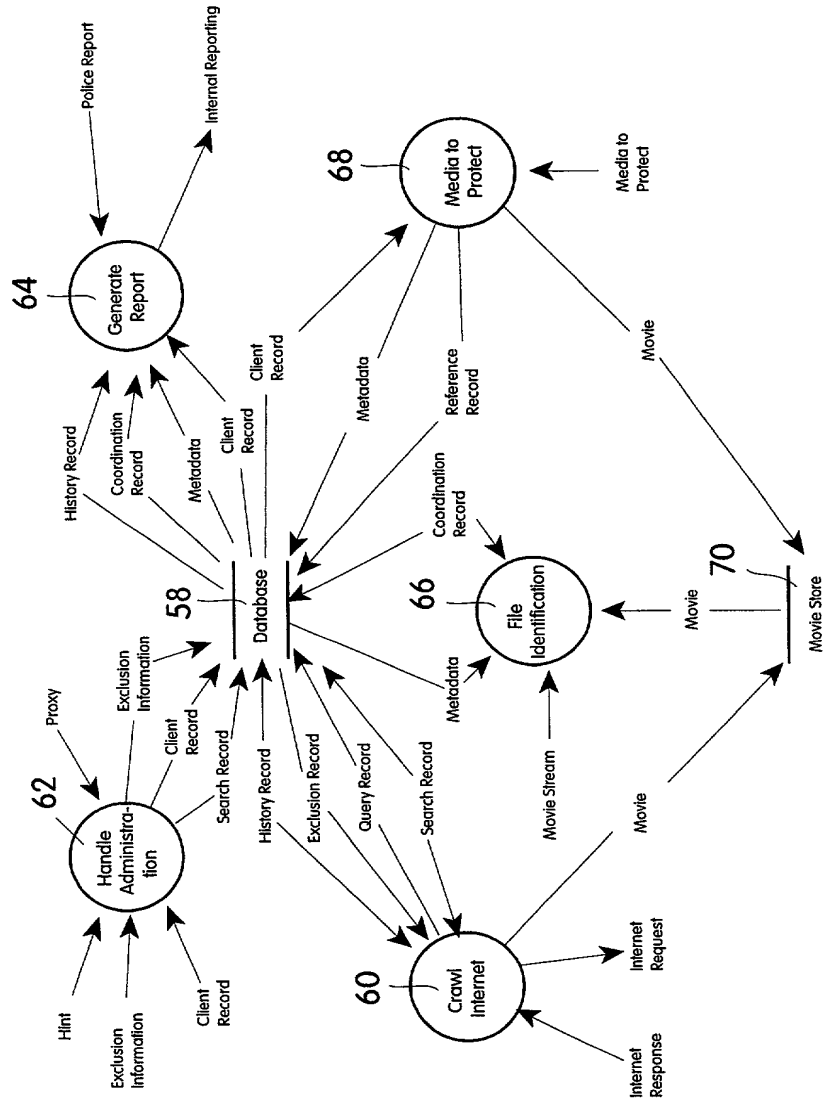


FIG. 9

FIG. 10



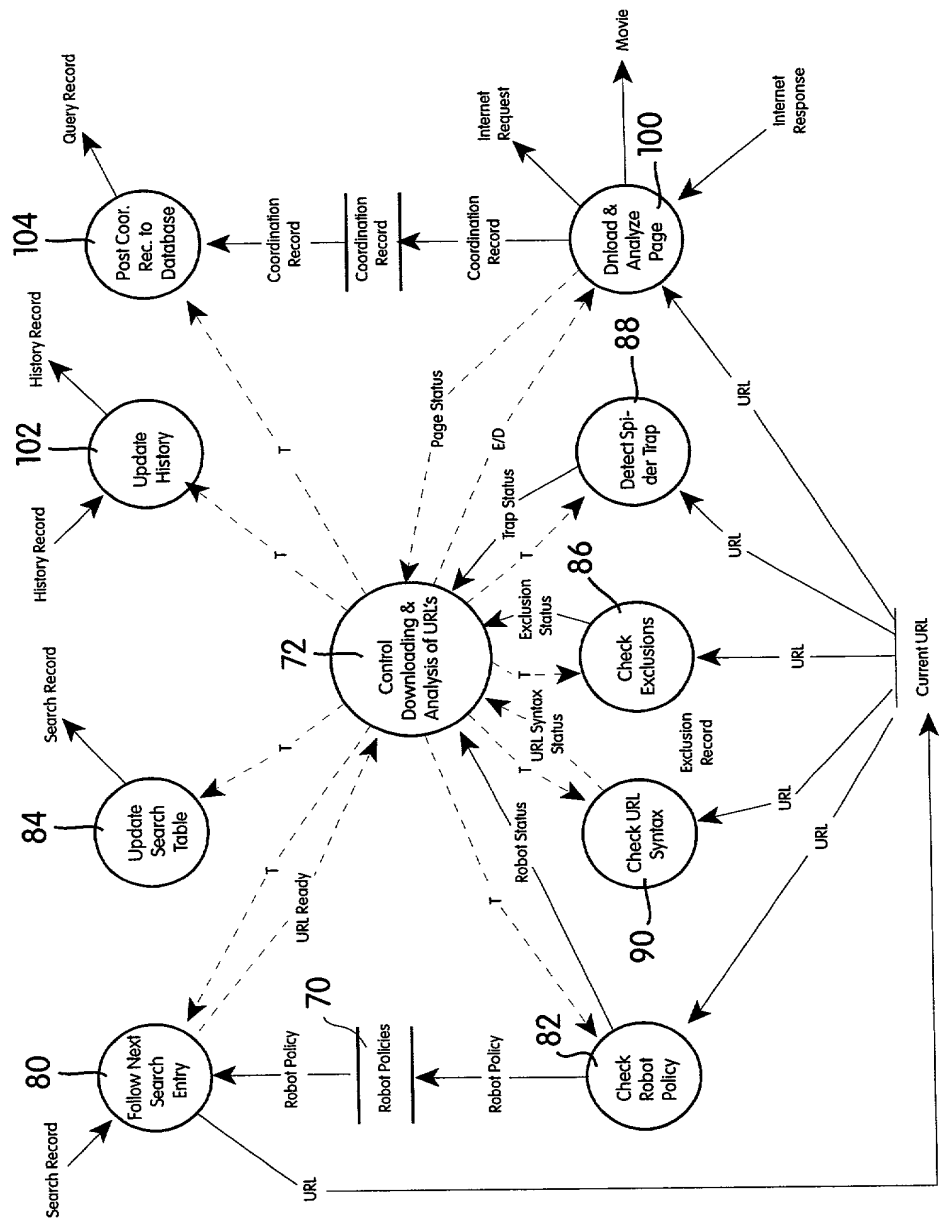


FIG. 11

```

graph TD
    Start([start]) -- "no URL" --> GetURL[Getting next URL]
    GetURL -- "URL ready" --> CheckSyntax[Validating URL]
    CheckSyntax -- "bad syntax" --> UpdateHistory1[T: "update History"]
    CheckSyntax -- "poor URL" --> CheckExclusions[Checking exclusion]
    CheckExclusions -- "excluded" --> UpdateHistory2[T: "update History"]
    CheckExclusions -- "not excluded" --> CheckRobotPolicy[Checking robot policy]
    CheckRobotPolicy -- "Robot false" --> UpdateHistory3[T: "update History"]
    CheckRobotPolicy -- "Robot true" --> DetectSpiderTrap[Checking for trap]
    DetectSpiderTrap -- "trap detected" --> UpdateHistory4[T: "update History"]
    DetectSpiderTrap -- "no trap detected" --> DownloadFirstPart[Downloading first part]
    DownloadFirstPart -- "can't download" --> UpdateHistory5[T: "update History"]
    DownloadFirstPart -- "downloading OK" --> CheckFileType[Checking file type]
    CheckFileType -- "File type unknown" --> UpdateHistory6[T: "update History"]
    CheckFileType -- "File type detected" --> DownloadRestOfFile[Downloading rest of file]
    DownloadRestOfFile -- "can't download movie" --> UpdateHistory7[T: "update History"]
    DownloadRestOfFile -- "Stream detected" --> PostCoordinationRecord[Post coordination record to database]
    PostCoordinationRecord -- "Stream detected" --> UpdateHistory8[T: "update History"]
    PostCoordinationRecord -- "Stream detected" --> DownloadAndAnalyzePage1[download and analyze page]
    PostCoordinationRecord -- "Stream detected" --> CheckHTMLForInterest[Checking HTML for interest]
    CheckHTMLForInterest -- "not interesting" --> UpdateHistory9[T: "update History"]
    CheckHTMLForInterest -- "interesting" --> UpdateSearchTable[T: "update search table"]
    UpdateSearchTable -- "interesting" --> DownloadAndAnalyzePage2[download and analyze page]
  
```

The flowchart illustrates the logic of a spider program. It begins with a 'start' node, leading to 'Getting next URL'. If there is 'no URL', it loops back to 'Getting next URL'. If a 'URL ready', it proceeds to 'Validating URL'. From 'Validating URL', if there is 'bad syntax', it leads to 'T: "update History"'. If there is a 'poor URL', it leads to 'Checking exclusion'. From 'Checking exclusion', if it is 'excluded', it leads to 'T: "update History"'. If it is 'not excluded', it leads to 'Checking robot policy'. From 'Checking robot policy', if the robot is 'false', it leads to 'T: "update History"'. If the robot is 'true', it leads to 'Checking for trap'. From 'Checking for trap', if a 'trap detected', it leads to 'T: "update History"'. If 'no trap detected', it leads to 'Downloading first part'. From 'Downloading first part', if it 'can't download', it leads to 'T: "update History"'. If 'downloading OK', it leads to 'Checking file type'. From 'Checking file type', if the 'File type unknown', it leads to 'T: "update History"'. If the 'File type detected', it leads to 'Downloading rest of file'. From 'Downloading rest of file', if it 'can't download movie', it leads to 'T: "update History"'. If a 'Stream detected', it leads to 'Post coordination record to database'. From 'Post coordination record to database', it leads to 'T: "update History"', 'download and analyze page', 'Checking HTML for interest', and 'update search table'. From 'Checking HTML for interest', if 'not interesting', it leads to 'T: "update History"'. If 'interesting', it leads to 'update search table'. From 'update search table', it leads to 'download and analyze page'.

FIG. 12

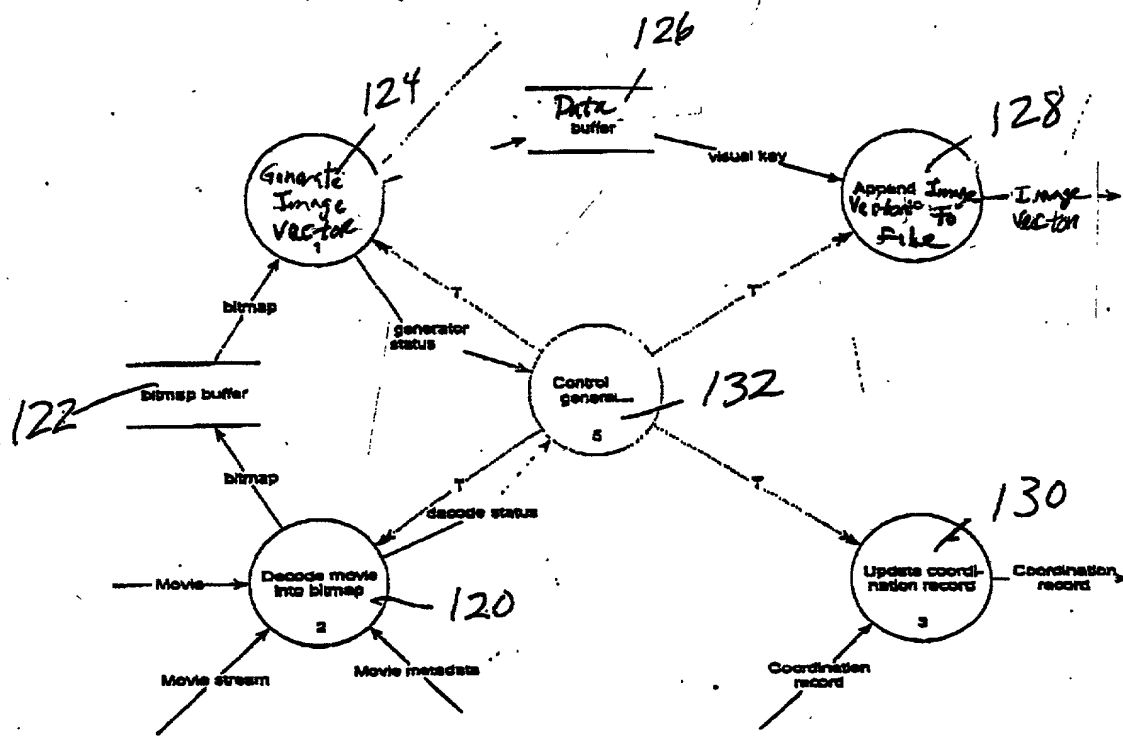


FIG. 13

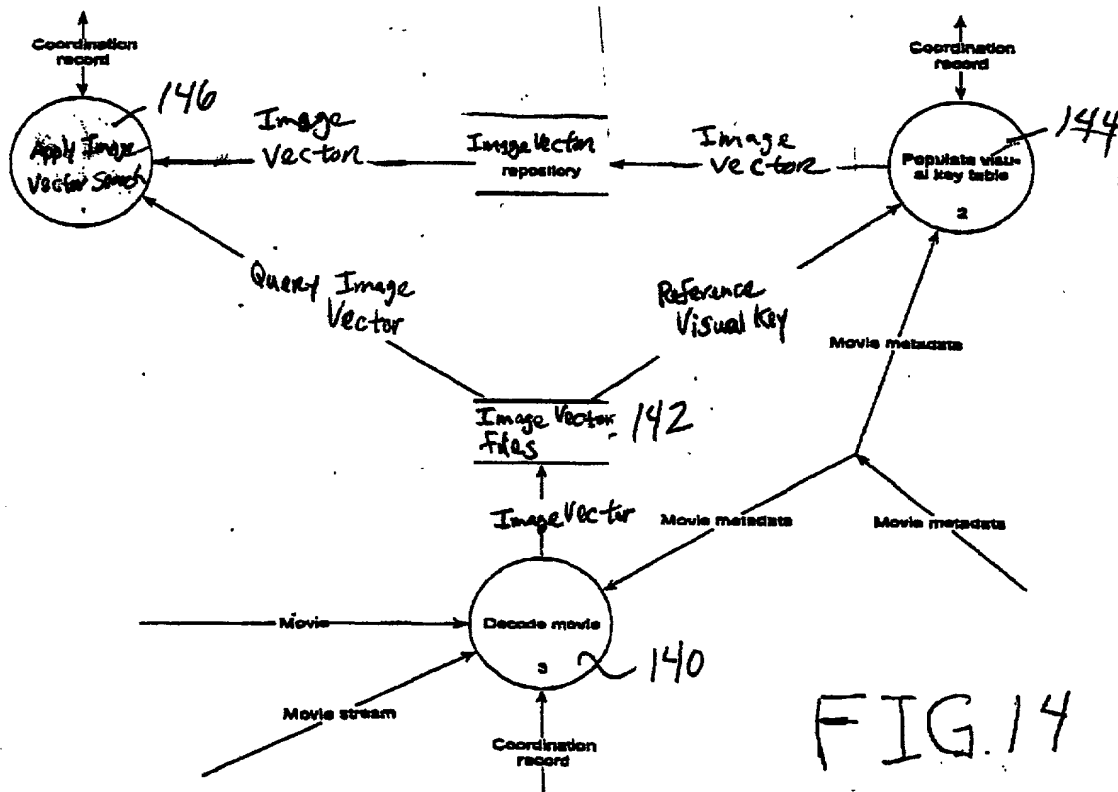


FIG. 14